REAL-TIME TRAFFIC IN DEFLECTION NETWORKS

Wladek Olesinski & Pawel Gburzynski
Department of Computing Science
University of Alberta
Edmonton, Alberta, CANADA T6G 2H1

Keywords: deflection networks, isochronous traffic, discrete simulation

ABSTRACT

We investigate experimentally the performance of deflection networks for jitter-sensitive traffic. It is a common belief that pure deflection networks are not suitable for isochronous applications, and networks based on a connection-oriented paradigm are the only solution in such cases. We argue to the contrary and present simulation results supporting our claims.

1 INTRODUCTION

By a *pure deflection network* we understand a switching network that never loses a packet at an intermediate switch because of a limited buffer space. Packets that cannot be relayed via their preferred routes (because those routes are busy) are *deflected* via suboptimal routes. This concept is traditionally illustrated by the Manhattan-street networks (MSN) introduced and analyzed in (Maxemchuk 85-93). No buffer space is required for storing packets that cannot be immediately relayed via their optimal routes, although it may make sense to store such packets temporarily before deflecting them right away (Maxemchuk 87).

Because of deflections, packets belonging to the same session may arrive at the destination out of order. This property of deflection networks has been traditionally perceived as a serious disadvantage compromising their reliability in applications, in which the timing of arriving packets is important (e.g., voice and video). This presumption has brought us the connection-oriented paradigm of ATM as the only solution to be adopted by the "serious" networks of the future. But, as one of the present authors has argued in (Baransel 95), switched networks are haunted by other problems that are absent in deflection networks. The proliferation of traffic classes in ATM and the obscure and inefficient ways in which ATM handles datagram traffic (Bae 91, Le Boudec 92) indicate that the connection-oriented approach is not perfect and that it hardly solves the problems of datagram-oriented networks (including deflection networks).

Because of the statistical multiplexing and unpredictability of real-life traffic scenarios, switched networks are bound to lose packets. In a deflection network, the problem of packet misordering is solved at the destinations by using *reassembly buffers* (Choudhury 91, Huang 95). A packet can be lost, if it happens to arrive so late that the reassembly buffer can no longer help. Thus we are confronting two approaches: one in which packets can be lost at all intermediate switches-relays because of the limited buffer space, and another in which packets can be lost at the destination for essentially the same reason. Thus, at this level of perception, there is no significant difference between the two approaches.

One advantage of deflection networks over store-and-forward networks is that the former don't try to "fix what ain't broke." In particular, datagram traffic requires no reassembly buffer at the destination (and no buffer space anywhere in the network). A destination in a deflection network allocates the reassembly space on a per session basis; thus, buffers are only used when required by the session. Clearly, the destination can "know better" how much buffer space is needed to accommodate its needs. Even if at first sight the amount of buffer space needed to accommodate an isochronous session appears somewhat large, one should keep in mind the simplicity and locality of the underlying allocation problems.

If we give up connection-orientedness as the prerequisite for networking, we will see that the number of sessions that actually need this paradigm will drop significantly. Let us point out that many traffic scenarios traditionally viewed as stream-oriented are only so because some old-fashioned protocols insist on viewing them that way. File transfer (FTP) is a good example. Note that the operating systems of the hosts involved in a file transfer perceive the file as a random collection of pages, which can be transferred independently in any order. Even better, the source could actually optimize the transfer by selecting the pages in the order that would minimize the total time needed to read them from the disk.

If we look carefully at those communication scenarios that appear to require the preservation of packet ordering, we will see that most of them fit into the following categories:

1. Scenarios that could be carried out with packets arriving in any order. They enforce packet ordering because some obsolete higher-level protocols view them as stream-oriented scenarios.
2. Scenarios involving relatively short transfers (e.g., a piece of text to appear on a screen). Messages of this

sort can be safely reassembled in a small buffer space at the destination.

3. Long, sustained, isochronous transfers that actually require packets to arrive in order (e.g., video, voice).

Note that the scenarios from the last category typically admit a non-zero packet loss rate. Consequently, one can implement them with a limited reassembly buffer, dropping packets that arrive out of sequence while the buffer is full. Quality of service requirements specified for such scenarios in store-and-forward networks include a non-zero (acceptable) packet loss rate.

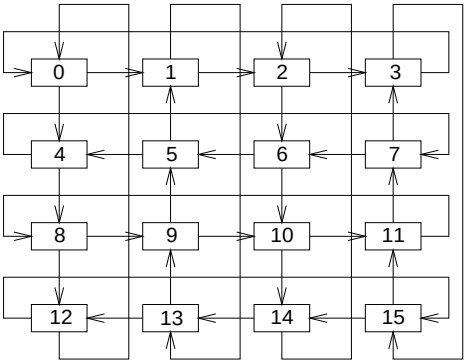
Similar to ATM networks, deflection networks are open for simple schemes aimed at improving the quality of service for some traffic patterns, based on traffic shaping at the source. Many people believe that in the face of long network delays and traffic unpredictability at intermediate switches, source policing is the only viable solution to the problems of congestion in statistically multiplexing networks.

In this paper we report the results of some simulation experiments aimed at determining the buffer space requirements in deflection networks for reassembling ordered streams of packets at their destinations. We use the *jitter* (understood as the standard deviation of packet delay at the destination expressed in slots) as the measure of disorder introduced by deflection operating under different conditions.

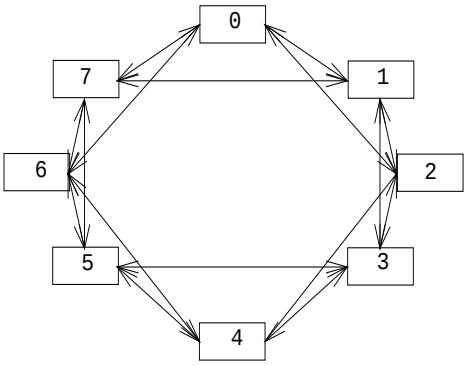
2 THE MODEL

We consider three deflection network topologies: torus, chordal ring, and triangle (Figure 1). The torus represents well-balanced networks with good reachability. The ring is still regular, but the reachability is poorer than in the torus. Finally, the triangle is a topologically biased network: different stations have different perceptions of their neighborhood. The networks operate in a slotted manner, in a way similar to MSN (Maxemchuk 87). In one routing cycle, the switch accepts incoming slots from all its input ports and routes them to the output ports. If an incoming slot is nonempty and addressed to the current switch, the switch receives the contents of the slot and marks it as empty. If an incoming slot is empty, the switch is free to fill it with its own outgoing packet. The routing decision assigns output ports to all incoming slots that appear nonempty after the above operations.

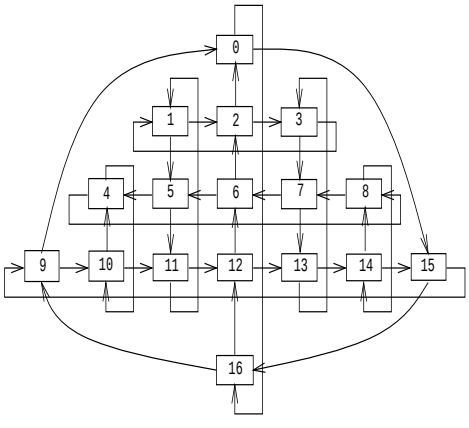
Let $d(S_1, S_2)$ denote the length of the shortest path from switch S_1 to S_2, expressed as the number of links (hops). Let $S(s)$ denote the destination switch of a nonempty slot s. Assume that switch S_a is making a routing decision for k nonempty slots routed in the current cycle. This decision is carried out in such a way as to minimize $P = \sum_{i=0}^{k-1} d(S_i, S(s_i))$, where s_i is a nonempty slot being assigned to an output port, and S_i is the immediate neighbor of S_a to which the slot is routed. We say that a routing scheme with this property is *locally optimal*. If several assignments of the outgoing slots to output ports produce the same minimum value of P, one of them is chosen at random.



(a) Torus, size 16, connectivity 2



(b) Ring, size 8, connectivity 4



(c) Triangle, size 17, connectivity 2

Figure 1: Network topologies.

This randomization of routing rules has been postulated in (Maxemchuk 91) for MSN as a means of avoiding *livelocks*.

In our model, we assume that slots are never buffered at a switch, except for the alignment and routing. Although generally a higher throughput can be achieved by using limited buffers, it was shown in (Greenberg 86) that for MSN $\lim_{N\to\infty} T(B)/C = 1$, where N is the number of switches in the network, B is the amount of extra buffer space at a switch, $T(B)$ is the observed throughput with the extra buffer space, and C is the network capacity, i.e., the maximum throughput reachable with infinite buffers. This spectacular result holds for any B, including 0. It is suspected that the same formula holds for all deflection networks with a reasonable topology and connectivity (Borgonovo 92). Thus the impact of the extra buffer space on the network performance asymptotically vanishes as the network becomes larger and larger.

Initially, we assume that the total delay involved in a single hop in the network is the same for all links and equal to a single slot. The impact of longer links is discussed in section 3.5. There are two selected switches: the source (S) and the destination (D). S sends its packets only to D; D receives packets only from S. We investigate the performance of the traffic session between the two selected switches under different background traffic scenarios in the remaining part of the network.

Each transmitting switch generates packets at some average rate expressed in $slots^{-1}$. The selected source generates packets at a fixed rate. Our objective is to monitor the standard deviation of the interarrival time between consecutive packets observed at the selected destination. Transmission rules for sources other than S depend on the selection of the background traffic.

As soon as a packet sent by S arrives at D, its delay and jitter are computed. The delay is measured from the moment when the packet's predecessor was received, and the jitter is the standard deviation of the delay. These measures are well defined because in our model (and intentionally in all pure deflection networks) packets are never dropped at intermediate switches. Note, however, that packet delay can take negative values, which happens when a packet and its predecessor have been misordered. No performance measures are calculated for the background traffic.

3 RESULTS

In this section we present some of our simulation results obtained for the three reference networks mentioned in the previous section. We considered networks of various sizes, but for the sake of brevity, we restrict our presentation to networks with $N = 256$ (torus and ring) and $N = 257$ (triangle) switches. Note that 257 is the closest approximation of 256 for which the triangle topology is complete. The connectivity of the torus and triangle (the number of link pairs per switch) is 2 and 4, whereas for the ring it is 4 and 8.

We have carried out experiments for several different locations of the selected source and destination. For the results presented in this paper, the two switches have been chosen as two most distant switches in the network (thus we are looking at the worst case). For the torus and ring (the topologically symmetric networks), any pair of antipodal switches has this property and all such pairs are equivalent. For the triangle, the sender is switch 17 (for $k = 2$) and 41 (for $k = 4$), and the recipient is switch 250 ($k = 2$) and 166 ($k = 4$).

3.1 Poisson Background Traffic

In this scenario, the background traffic is uniform, with all sources and destinations (except S and D that don't contribute to the background traffic) being equally probable. Every switch generates packets according to the Poisson distribution with a given mean. The load of the network (the horizontal axis in the performance graphs) determines how many new packets appear in the network within one slot time unit. Packets that cannot be expedited immediately are stored in queues at their source switches.

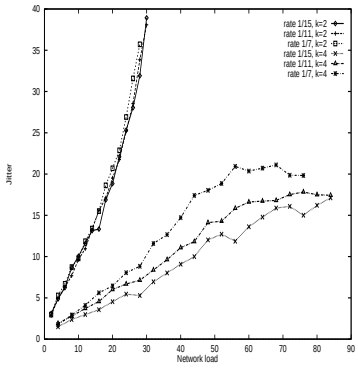
Figure 2 shows the jitter vs background load for different transmission rates of the selected source S and different connectivities (k). As expected, the jitter increases with the increasing background load. Also, the transmission rate of the source has a visible impact on the jitter, especially in the torus, which phenomenon becomes more pronounced for higher-connectivity networks. This is somewhat surprising at first sight, because one would expect that higher-connectivity/reachability networks should offer not only lower jitter, but also better stability. It seems that sometimes the abundance of alternative routes with slightly varying shortest-path costs is not an advantageous property from the viewpoint of containing the jitter.

Nonetheless, although low connectivity/reachability networks sometimes offer better stability, they consistently incur significantly higher jitter than their better connected counterparts, even if the background traffic is not bursty and relatively smooth. Note that the connectivity-2 torus network is in fact the standard architecture of MSN.

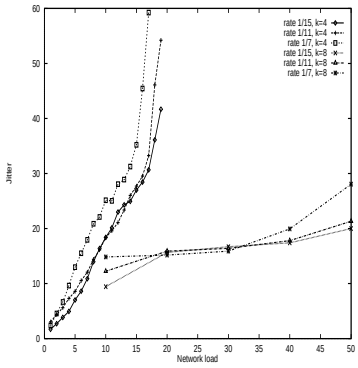
As the network becomes larger, the jitter seems to grow proportionally to the increasing load (a larger network can carry more load). This phenomenon results from two factors: the increased distance between the two peers involved in the monitored session, and the increased opportunities for deflected packets to wander through remote regions of the network. Experiments indicate that the jitter depends primarily on the distance and increases only slightly with the network size, if the distance and (normalized) load in the neighborhood remain fixed.

3.2 Correlated Poisson Background Traffic

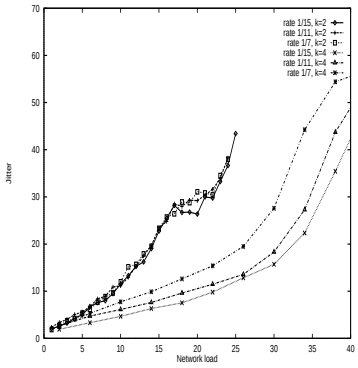
This time, the uniformity of the Poisson traffic model is disrupted by two additional parameters l (session length) and P (correlation level). Every l slots, a set of source-destination pairs is selected at random, with the number of pairs $n = P \times \frac{N}{2}$. For the next l slots, each of the selected senders will be generating uniform Poisson traffic addressed to its one dedicated destination. The remaining background switches will carry on as before, generating Poisson traffic to



(a) Torus, N=256



(b) Ring, N=256



(c) Triangle, N=257

Figure 2: Uniform Poisson background traffic in networks of different sizes and topologies.

randomly chosen destinations from outside of the selected set.

The intention of this traffic model is to represent scenarios in which there is a sustained traffic of a non-trivial duration between pairs of switches engaged in some correlated sessions. The background load of the network can still be characterized by the global arrival rate of background packets.

The results of these experiments are very close to the previous ones for practically all ranges of l and P (see Figure 3).

This indicates that deflection networks are not very sensitive to the changing patterns of loads, at least for as long as the global load in the neighborhood remains at the same level. This property is advantageous for large networks, because their global loads tend to change slowly, although there may be local bursts of correlated activities.

3.3 Bursty Synchronous Background Traffic

The intention of this scenario was to model a situation in which the background switches generate independent bursts of traffic transmitted at the highest possible rate. Intuitively, such a scenario should be malicious for the monitored synchronous session, because the interference is heavy and essentially non-deterministic.

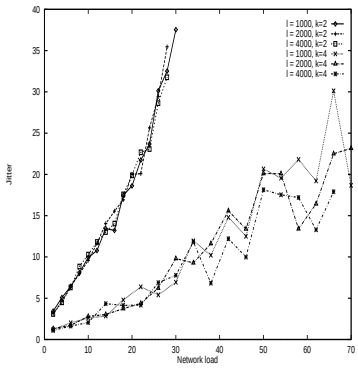
The traffic generator at every background switch operates as follows. It sleeps for an exponentially distributed random amount of time with mean t. After that time, it generates a burst of packets, its population determined by an exponentially distributed random number with mean n. All these packets are sent at the highest available rate. The intensity of the background traffic is adjusted by modifying n, i.e., more intense traffic means longer bursts rather than a shorter interarrival time between bursts.

The results for this setup are shown in Figure 4. Notably, the observed jitter is not worse than for the other scenarios, except for the large ring, which seems to be most heavily penalized by the nondeterminism and burstiness of the background traffic. Note that large rings are not well suited for deflection because of the large deflection penalty, which increases rapidly with the increasing network size.

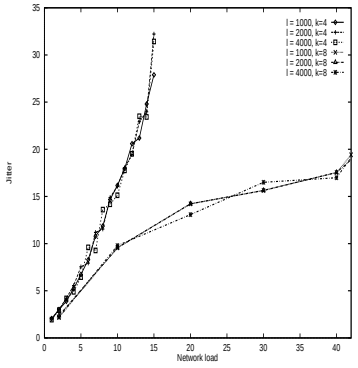
3.4 Other Traffic Patterns

We have investigated other background traffic scenarios, including long sustained synchronous sessions, self-similar bursty patterns, and bursty correlated patterns, as well as mixes of different traffic types. None of those experiments produced results that would be worse than those mentioned in the preceding sections.

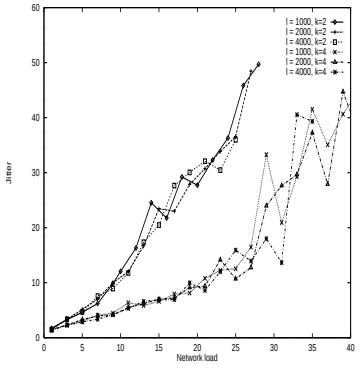
Generally, for the increasing network size, when the contribution of an individual session counts less in the total statistical mix, the observed jitter for the monitored synchronous session tends to remain low and acceptable. Of course one can devise artificial sessions especially designed to interfere in the worst possible way with the monitored session, but one can do the same for store-and-forward networks, including ATM. Therefore, we were not interested in



(a) Torus, N=256



(b) Ring N=256



(c) Triangle N=257

Figure 3: Correlated Poisson background traffic in networks of different sizes and topologies. $P = 0.4$, rate of S is 1/14, $n = 51$ (in larger network) or $n = 12$ (in smaller network).

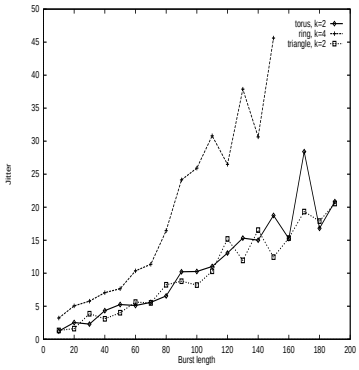


Figure 4: Bursty synchronous background traffic in networks of all three topologies ($t = 20$).

demonstrating that statistical fluctuations may produce arbitrarily large deviations in the observed hop count (which is obvious and trivial), but rather in determining how easy it is to stumble upon such a configuration in a mix of background sessions with some statistical and reasonably unfriendly properties.

It turns out that it isn't easy to produce really bad scenarios by accident, or even intentionally. In particular, bursty background traffic scenarios are generally much less unfriendly than one would expect. If the bursts are short, then the background traffic appears relatively smooth and, although the monitored synchronous session is disturbed, this disturbance is not long enough to worsen the jitter significantly. On the other hand, a long burst may disturb a few packets at the beginning, but later, while it persists, the network will learn to circumvent it and the disruption will cease to affect the jitter in a significant way.

3.5 Larger Propagation Delays

So far, we have assumed that propagation delay expressed in slots is low and equal to one slot. One can argue that in a high-speed network of a non-trivial size, the delay suffered by a packet on its single hop may be considerably longer. This may increase the jitter and buffer requirements at the destination.

Intuitively, if the delay and jitter are normalized to hops, their actual values can be obtained by multiplying the normalized values by the (average) channel length expressed as the propagation time.

We have performed simulations for the propagation delays between neighboring switches equal to $l = 1, 5, 10, 15$ slots. If we assume, e.g., as in (Choudhury 91), that the link capacity is 150 Mb/s, the slot length is 53 bytes, and the speed of signals in the medium is $0.69c$, then there are approximately 1.7 slots in transit on each kilometer of the link. It means that the propagation delay of 1 slot per link corresponds roughly to a network diameter of 10 km (assuming a torus grid). By the diameter of a mesh network we mean the maximum length of the shortest path between

a pair of switches. The propagation delays of 5, 10, and 15 slots correspond to MANs of diameters 50, 100, and 150 km, respectively.

The simulation results indicate that, as expected, the jitter increases in proportion to the link length, with the processing delay at a switch contributing a constant factor.

3.6 Reassembly Buffers

Typically, the quality of service requirements of a synchronous session are specified in terms of packet loss rate rather than jitter. From the viewpoint of measurement, the jitter is a more convenient parameter to investigate, because under normal conditions not too many packets are lost, and it may take a long time to collect a meaningful statistical sample. It turns out, however, that the jitter can give us a fairly good idea of the packet loss rate for a given size of the reassembly buffer.

Our reception model operates as follows. Every packet arriving at the destination is first inserted in the reassembly buffer of size R. As soon as the number of packets in the buffer reaches $L = F \times R$ $(0 < F < 1)$, the destination starts to "receive" packets (i.e., extract them from the buffer) at the rate equal to the transmission rate. The destination tries to receive the packets in the same order in which they have been sent. If a packet is not available when its turn comes, it is marked as lost. When that packet arrives later from the network it will be discarded. Similarly, a packet arriving while the reassembly buffer is full is dropped immediately. The role of L is to provide an initial backlog of packets to be received—to compensate for the variability and unpredictability of network delays.

At first sight, it would seem that F should be set to 0.5, with one half of the buffer space used to compensate for the unpredictability of delays in the "late" direction, and the other half used to buffer packets arriving early. Our experiments consistently indicate that the best value of F (for all deflection networks and traffic patterns) is about 0.85. This means that it is more important to account for those packets that have been delayed in the network than for those that may find the reassembly buffer full upon their arrival.

Numerous simulation results indicate that for R equal twice the observed jitter, practically no packets are ever lost (the loss rate is statistically unmeasurable and below any sensible QoS requirements for synchronous traffic). The loss rate of 1% typically occurs for R equal to the jitter and drops very fast as more buffer space is added. For a given percentage of packet loss, the buffer size is a linear function of the jitter.

4 SUMMARY

We have presented some experimental results hinting at the expected jitter (and consequently buffer space requirements) in deflection networks used to carry traffic with timing constraints. Although these results are preliminary and more studies are needed to determine specific design criteria for real-life deflection networks, we find them optimistic and encouraging. For example, consider a "large" torus network with the average propagation distance between a pair of neighboring switches equal to 10 slots. Assuming the transmission rate of 150 Mb/s, this translates into the network diameter of 100 km. With the reassembly buffer size of 200, the network can cater to practically any sensible isochronous session, offering a very low packet loss rate. This buffer space is rather small (considering that the packet size of 53 bytes corresponds to the ATM cell) and allocated on a per session basis exclusively at the destination. Also, it has been arrived at under the assumption that the isochronous traffic receives no special treatment anywhere in the network.

REFERENCES

Bae, J., and T. Suda. Survey of traffic control schemes and protocols in atm networks. *Proceedings of the IEEE*, 79:170–189, 1991.

Baransel, C., W. Dobosiewicz, and P. Gburzynski. Routing in multi-hop switching networks: Gbps challenge. *IEEE Network Magazine*, (3):38–61, 1995.

Borgonovo, F., and L. Fratta. Deflection networks: architectures for metropolitan and wide area networks. *Computer Networks and ISDN Systems*, (24):171–183, 1992.

Le Boudec, J-Y. The asynchronous transfer mode: a tutorial. *Computer Networks and ISDN Systems*, 24:279–309, 1992.

Choudhury, A., and N. Maxemchuk. Effect of a finite reassembly buffer on the performance of deflection routing. *Conference Record of the International Conference on Communications (ICC)*, 3:1637–1646, 1991.

Greenberg, A., and J. Goodman. Sharp approximate models of adaptive routing in mesh networks. In O. Boxma, J. Cohen, and H. Tijms, editors, *Teletraffic Analysis and Computer Performance Evaluation*, pages 255–270. Elsevier Science Publishers B.V. (North-Holland), 1986.

Huang, H-Y., T. Robertazzi, and A. Lazar. A comparison of information based deflection strategies. *Computer Networks and ISDN Systems*, 27:1399–1407, 1995.

Maxemchuk, N. The Manhattan street network. In *Proceedings of GLOBECOM'85*, pages 255–261, 1985.

Maxemchuk, N. Routing in the Manhattan Street Network. *IEEE Transactions on Communications*, 35(5):503–512, May 1987.

Maxemchuk, N. Comparison of deflection and store-and-forward techniques in Manhattan-street network and shuffle-exchange networks. In *Proceedings of IEEE INFOCOM'89*, pages 800–809, 1989.

Maxemchuk, N. Problems arising from deflection routing. In Pugolle, editor, *High Capacity Local and Metropolitan Networks*, pages 209–233. Springer Verlag, 1991.

Maxemchuk, N., and R. Krishnan. A comparison of linear and mesh topologies–DQDB and the manhattan street network. *IEEE Journal on Selected Areas in Communications*, 11(8):1278–1301, Oct. 1993.